



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

XJ

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/115,359	07/14/1998	JOHN W. MERILL	INTL-0038-US	1288
7590	03/01/2005		EXAMINER	
TIMOTHY N TROP TROP PRUNER HU , P.C. 8554 KATY FREEWAY STE 100 HOUSTON; TX 77024			KNEPPER, DAVID D	
			ART UNIT	PAPER NUMBER
			2654	
DATE MAILED: 03/01/2005				

Please find below and/or attached an Office communication concerning this application or proceeding.



UNITED STATES PATENT AND TRADEMARK OFFICE

---

COMMISSIONER FOR PATENTS  
UNITED STATES PATENT AND TRADEMARK OFFICE  
P.O. Box 1450  
ALEXANDRIA, VA 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

**MAILED**

**MAR 01 2005**

**Technology Center 2600**

**BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES**

Application Number: 09/115,359

Filing Date: July 14, 1998

Appellant(s): MERILL, JOHN W.

---

Timothy N. Trop  
For Appellant

**EXAMINER'S ANSWER**

This is in response to the appeal brief filed 16 July 2004.

**(1) Real Party in Interest**

A statement identifying the real party in interest is contained in the brief.

**(2) *Related Appeals and Interferences***

A statement identifying the related appeals and interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief.

**(3) *Status of Claims***

The statement of the status of the claims contained in the brief is correct.

**(4) *Status of Amendments After Final***

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

**(5) *Summary of Invention***

The summary of invention contained in the brief is deficient because the summary provides a description which does not correspond to the claimed invention.

A correct summary of the invention may be found on page 2 of the specification repeated below for convenience:

“In accordance with one aspect, a method for recognizing speech includes providing a speech engine with a vocabulary of command sets. The appropriate command set for the current software application is communicated to the speech engine.

In accordance with another aspect, a method of recognizing speech includes associating speech units with an identifier. The identifier is also associated with an action to be taken in response to the speech unit. The identifier for a given spoken speech unit is determined and the identifier is provided to a software object.”

As noted on page 3 of the specification, lines 16-20: "The application 10 may respond either to spoken commands or tactile inputs. Tactile inputs could include pushing a keyboard key, touching a display screen, or mouse clicking on a visual interface."

The claimed invention does not correspond to the Summary presented by the applicant. For example, the claims do not indicate any use of grammar, navigator function, GetControlInfo, IoleControl, state errors, accelerator key, "jump" command, "off" command, nor is any claim language directed towards invoking particular functions from any window or frame reference. There are too many improper references in the applicant's Summary to specifically point out but it is clear that this Summary is a far cry from the claimed invention.

### *Summary of the State of the Art*

The applicant correctly identifies some background information in the specification on pages 1-2. The applicant's description of API (Application Program Interface) is correct. It is known that parallel processing can produce multiple sources of inputs which need to be synchronized in order to be properly handled, for example, by an operating system which would control the overall functions of a computer. An API may handle multiple sources of input.

### **(6) Issues**

The appellant's statement of the issues in the brief is substantially correct. The changes are as follows: Claims 33-52 are rejected under 35 USC 102(e) in view of Trower(5,983,190) as stated by the Appellant.

Claims 33-52 are also rejected under 35 USC 103 in view of Hashimoto (5,632,002) which was not stated by the Appellant. Hashimoto was applied in a previous rejection. The

instant claims are broader and do not contain limitations addressed by additional art previously decided by the Board so the arguments apply similarly to Hashimoto.

**(7) *Grouping of Claims***

The rejection of claims 33-52 stand or fall together because appellant's brief does not include a statement that this grouping of claims does not stand or fall together and reasons in support thereof. See 37 CFR 1.192(c)(7).

**(8) *ClaimsAppealed***

The copy of the appealed claims contained in the Appendix to the brief is correct.

**(9) *Prior Art of Record***

5,893,190	Trower, II et al.	9 Nov 1999
5,632,002	Hashimoto et al.	20 May 1997

**(10) *Grounds of Rejection***

The following ground(s) of rejection are applicable to the appealed claims:

1. Claims 33-52 are rejected under 35 USC 102(e) as being anticipated by Trower (5,983,190).

As per claims 33, 39, 45, 46: "providing a single software object that receives spoken and non-spoken command information" (his collection of commands that an agent object will respond to when a client becomes active and also enables its selection through speech recognition, col. 27, lines 5-26); and

"firing an event when an object receives spoken command information" (his OLE object can expose a set of functions that is derived from IDispatch and includes method and property access functions that another program can call directly. This is sometimes called a 'dual' interface because other programs can invoke an object's methods through the IDispatch interface and directly

through this second type of interface, col. 21, lines 17-23 – This clearly teaches a single object may perform multiple functions, can invoke multiple methods and can include multiple interfaces to receive different types of information).

Firing the same event when receiving spoken and “non-spoken command” is clearly included by Trower who will allow his object’s methods to be controlled by more than one interface. Thus, it is apparent that any known control methods could invoke any known program functions using the interface taught by Trower. Since Trower explicitly teaches keyboard and pointing devices (figure 1) as well as speech recognition (figure 3), any combination of these is anticipated.

Claims 34, 40, 48: “Providing a speech engine with a vocabulary command set for at least two tasks” (his speech recognition engine analyzes digitized audio input to identify words or phrases, col. 2, lines 47-48.

Claims 35, 41, 49: “Associating a command with an identifier” is taught in column 2 where he states that clients can specify input commands including both speech and cursor device input for the character (lines 29-30) and Clients specify the speech or cursor input that a character will respond to . . . The server monitors input from the operating system (cursor device input) and the speech recognition engine (speech input) for this input (lines 63-66). Thus, he clearly anticipates the use of a common identifier for both speech and cursor device inputs.

“Associating the identifiers with actions to be taken in response to a command” is taught by his ability to play animation and speech output to animate the character . . . (col. 2, lines 31-34). Thus, he clearly anticipates the use of externally visible actions as opposed to actions internal to the computer that the user would not see. However, it is noted that the claim language does not state what the action is and would therefore be anticipated even if these other actions were absent.

Claims 36, 42, 50: “Instantiating an object in a container” is taught by figure 10 and column 20, lines 1-19 and 55-62 where he teaches that the server must first create the object (i.e., instantiate an object of a class supported by the server application). “Communicating the identifier to the object for a command” is inherent for any known use of speech recognition to affect control of an object and is anticipated by any one of his use of input commands, character, name or text string in column 2 (for example) which affect his use of commands that an agent object will respond to when a client becomes active . . . The client can also set the Voice property for a command, which enables its selection through speech recognition (col. 27, lines 5-26). Identifiers may also be embedded in an HTML page and example methods include Play, GestureAt, MoveTo, Stop, and Speak (col. 22, line 40 – col. 23, line 12), which give examples of specific actions that can be performed by the agent object.

Claims 37, 43, 51: “Communicating information about a first spoken command to the container, checking an active vocabulary list in the container to determine if the first spoken command is one used in an active task, and if the first spoken command is one used in an active task, transferring the identifier for the spoken command to the object.” This is taught by his speech recognition engine in communication with an audio input device for receiving speech input from the user . . . to identify the speech input commands; and in communication with the receiver for sending notification messages to the server when the speech input commands are detected (col. 39, lines 27-33).

See also his active vocabulary of the speech recognition engine (col. 28, lines 49-50), a list of commands that are currently available to the user (col. 27, lines 7-8), the client specifies the string value corresponding to the words or phrase to be used by the speech engine to recognize this command (col. 27, lines 57-59) and it is ultimately the end user that is controlling which client has the chance to become active (col. 33, lines 60-61).

Claims 38, 44, 47, 52: “Handling spoken and non-spoken commands in the same way” is taught by Trower as previously explained regarding claim 33 above because he clearly teaches that multiple inputs may activate the same object and it’s encapsulated functionalities. The same command would cause the same result whether spoken, typed, etc.

2. Claims 33-52 are rejected under 35 USC 103(a) as being unpatentable over Hashimoto (5,632,002)

Claim 33, 39, 45, 46: “providing a single software object” (his plurality of application programs 2 are operated in parallel, each application program 2 can exchange data such as the recognition vocabulary and the recognition result with the speech recognition system 1 . . . so that the speech input can be provided as the data input means for all the application programs 2 just as the other data input means such as the keyboard and the mouse, col. 18, lines 10-22);

“object . . . that receives both spoken and non-spoken command information; firing an event when said object receives spoken command information; and . . . unspoken command information” (his SIM 104 converts the speech inputs into the form acceptable by the GAP 103 such as that of the mouse inputs or the keyboard inputs, col. 59, lines 49-58 and the SIM 104 transmits the messages identical to those generated at the time of the operation command inputs by the usual input devices such as the keyboard and the mouse, col. 60, lines 1-21).

It is noted that Hashimoto does not explicitly teach “firing an event” related to a command. However, Hashimoto teaches that many different application programs 2 (figure 6) may be implemented with his interface. It would have been obvious to one of ordinary skill in the art that

even though Hashimoto does not describe specific actions to be employed by the applications that applications will perform actions desired by the users thereof. Performing only a desirable “event” is further suggested by Hashimoto in col. 11, lines 44-59 where he explains that his interface will limit recognition in order to avoid wasteful matching processing with respect to the unnecessary recognition vocabularies. This teaches that only desirable applications will be utilized and one of ordinary skill in the art would expect this to result in only events taking place as desired by the user by such a limitation to active vocabularies.

Claim 34, 40, and 48: “Providing a speech engine with a vocabulary command set for at least two tasks” (his ability to handle a plurality of application programs simultaneously, abstract – see also the specific teaching above to operate in parallel).

Claims 35, 41, and 49: “Associating a command with an identifier” (he teaches that such an operation command transmission can be easily implemented by utilizing the functions provided in the library of the window system. In the actual window system, there are cases in which the destination of the messages is not the GAP 103 itself but the object such as the window generated by the GAP 103, col. 60, lines 21-31).

Claims 36, 42, 50: “Instantiating an object in a container” is taught by figure 62 where he teaches initial set up ... 6101 ... concerning speech output processing 6102 followed by “Communicating the identifier to the object for a command” where he follows by speech input & speech output processing 6103 which would need to interpret the command to generate proper output based on certain input.

Claims 37, 43, 51: “Communicating information about a first spoken command to the container, checking an active vocabulary list in the container to determine if the first spoken command

is one used in an active task, and if the first spoken command is one used in an active task, transferring the identifier for the spoken command to the object” is taught by his program management table 13 has entries for the program Ids, the input masks, the recognition vocabulary lists, and the speech input flags . . . The speech input flag is a flag indicating whether or not the speech focus is focused on a corresponding one of the application programs 2 or not (col. 10, line 58 – col. 11, line 10).

Claims 38, 44, 47, 52: “Handling spoken and non-spoken commands in the same way” is taught as noted under claim 33 above because he has a single message processing unit MPU that is able to communicate with speech recognition and the application programs.

#### **(11) *Response to Argument***

It is noted that no argument was made by the applicant when new claims (33-52) were submitted on 23 Feb 2004 but the following response by the Examiner attempted to address the interpretation of “object”.

##### ***Response to Applicant’s New Claims submitted 23 Feb 2004***

The claims are now more clearly worded so that they more clearly read upon the prior art.

The argument that the prior art fails to teach a single software object is not convincing. One of ordinary skill in the art would know that software objects could contain simple programming of only one function or very complicated programming with multiple functions. Similarly, the definition of “object” that the applicant provided to the Board clearly states object: 2. In object-oriented programming, a variable comprising both routines and data that is treated as a discrete entity. Thus, even the definition provided by the applicant clearly acknowledges that a single object may contain multiple routines that may be treated as a discrete entity.

The rejections were re-worded to more clearly address the changes as reflected in the newly submitted claims. However, the grounds of rejection are essentially the same as those previously adjudicated by the Board. It is the Examiner's position that Res judicata applies now that the "factual question" regarding the "definition of an object" has been resolved (this was noted by the Board in paper 19 of 27 Oct 2003 in which the Board denied the applicant's request to make changes to their previous decision of paper 17 of 29 Jan 2004 in which the rejection was Affirmed).

***Response to Argument in the Appeal Brief (received 16 Jul 2004)***

On page 8 of the instant Appeal, the applicant continues to disagree that the definition of "object" was not argued during (or prior to) the previous Appeal. The Board properly declined to consider the applicant's new arguments which the applicant stated were based upon the definition of the term "object".

On page 8 of the instant Appeal, the applicant's statement that the definition was included in amended claims is clearly false. The text of the definition is quoted by the applicant and by the Examiner (underlined above) and this text does not appear in the claims. However, the definition is well-known and the term "software object" is used in a manner which is common to the art. The prior art relied upon uses the term "object" in the same manner as the applicant. In fact, the rejections have consistently used this term and have quoted (underlined text) such use by the prior art as the basis for the rejections.

The Appellant's statement on page 9 that the Board's decision effectively concurs with the Appellant's statement is also false. The Board's position is correct that a computer causes the functionality to be implemented in the manner desired by the appellants. The appellant fails

to address how the method differs nor how any apparatus other than a computer can perform the desired method. In fact, the definition relied upon for “object” is limited to a form of computer programming and, therefore, the Board’s position does not support the appellant.

It should be noted that the Examiner agrees with the Board’s statements that are quoted by the appellant on page 9 because they are technically accurate but do not precisely follow (in exacting terms) the rejections as stated by the Examiner. The Examiner’s position is, and has always been that the claimed software “object” which allows flexible input (i.e. – from speech or non-speech) is taught by Trower’s agent object (implemented according to standard OLE object programming) or by Hashimoto’s SIM 104 (Speech Interface Management which uses a window system).

Trower’s use of the term agent “object” is clearly for the same purpose as claimed by the appellant. Therefore, the definition of “object” is not a substantive issue. The fact that the definition relied upon by the appellant is from a standard dictionary of computer terms is considered evidence that the term and related software for providing such functionality is well known in the art.

It is noted by the Examiner that appellant has chosen not to traverse the rejection under 35 USC 103 to Hashimoto and therefore this rejection is proper based on the facts previously stated. However, assuming arguendo that the 103 rejection to Hashimoto is at issue the following comment is made.

Hashimoto used the abbreviation SIM for speech interface management system (col. 59, line 25 and figure 96). It is clear that the SIM represents a single element which is a computer program that also allows flexible input from speech, mouse and keyboard. Thus, even though

the terminology used by Hashimoto does not use the term "object" it is obvious that it is a functional equivalent as claimed by the appellant.

The appellant's characterization on page 10 that in the prior art "the information from one type of input goes to one type of control and the information from the other type of input goes to a different type of control" is contradicted by the teachings of the prior art itself which allows a single control to be programmed to deal with multiple input sources.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,



David D. Knepper  
Primary Examiner  
Art Unit 2654

February 16, 2005

Conferees:

Richemond Dorvil

David Ometz

Vijay Chawan



TIMOTHY N TROP  
TROP PRUNER HU AND MILES  
8550 KATY FREEWAY  
STE 128  
HOUSTON, TX 77024